Online Academic Course Performance Prediction using Relational Graph Convolutional Neural Network

Hamid Karimi\textsuperscript{1*}, Tyler Derr\textsuperscript{1*}, Jiangtao Huang\textsuperscript{2}, Jiliang Tang\textsuperscript{1}

\textsuperscript{1} Michigan State University, \{karimiha, derrtyle, tangjili\}@msu.edu
\textsuperscript{2} Nanning Normal University, China, hjt@gxtc.edu.cn

ABSTRACT

Online learning has attracted a large number of participants and is increasingly becoming very popular. However, the completion rates for online learning are notoriously low. Further, unlike traditional education systems, teachers, if any, are unable to comprehensively evaluate the learning gain of each student through the online learning platform. Hence, we need to have an effective framework for evaluating students’ performance in online education systems and to predict their expected outcomes and associated early failures. To this end, we introduce Deep Online Performance Evaluation (DOPE), which first models the student course relations in an online system as a knowledge graph, then utilizes an advanced graph neural network to extract course and student embeddings, harnesses a recurrent neural network to encode the system’s temporal student behavioral data, and ultimately predicts a student’s performance in a given course. Comprehensive experiments on six online courses verify the effectiveness of DOPE across multiple settings against representative baseline methods. Furthermore, we perform ablation feature analysis on the student behavioral features to better understand the inner workings of DOPE. The code and data are available from https://github.com/hamidkarimi/dope.

Keywords

Online courses, Student behavior modeling, Knowledge graph, MOOC, Graph neural networks

1. INTRODUCTION

Online learning has higher dropout and failure rates than traditional education systems. For instance, the completion rates of Massive Open Online Courses (MOOCs), an extension of online learning technologies, are low (0.7%-52.1%, with a median value of 12.6%, reported by \cite{20}). We also see similar situations in other online courses from universities such as Open University in the UK and China \cite{19}.

*Equal contribution and co-first author

Figure 1: Visual comparison of the learning/intervention process between online and in-person education systems.

Furthermore, since students typically drop out early in the courses \cite{33}, the platform is desired to detect which student is likely to drop out (or fail) as early as possible to intervene and to hopefully prevent these negative outcomes. Then the question is how we can assess students’ performance and detect those who are likely to drop out or fail in an online course. To answer this question, we first need to take a closer look at the online learning system and see how it differs from traditional learning.

As illustrated in Figure 1 (right side), in the traditional learning setting, instructors can interact with students, assess their performance, and take action to provide intervention if they sense a student is likely to perform poorly in the class. In online learning systems, however, the students primarily interact with the online platform, so we face a setting depicted in the left side of Figure 1. In this setting, there is inherently less interaction between students and instructors. More specifically, due to the high student-teacher ratio, teachers, if any, in the online learning systems are unable to comprehensively evaluate the learning gain of each student. Thus, we seek to develop a methodology that can harness the interactions of students with an online platform and accurately predict the course outcome (e.g., pass or fail). Such a system could then be used in real-time throughout the course to identify the students who are predicted to perform poorly and provide some intervention to them with the limited resources that are inherent in online systems.

Given the above discussion, we propose a framework named Deep Online Performance Evaluation (DOPE) to predict students’ course performance in online learning. DOPE first models the student course relations of the online system as a knowledge graph. To incorporate an aggregated overview of
1. We propose the use of a knowledge graph to model complex online learning environments to allow more rich data to be extracted as compared to representing the data in a traditional unstructured way; and

2. Our proposed framework to predict student course outcomes contains two novel components, namely a relational graph neural network to extract student and course embeddings from the formed knowledge graph and a recurrent neural network model for encoding student behavioral data according to their clicks in the online system.

2. PROBLEM STATEMENT

Suppose from the set of courses in an online system we have a subset of $m$ courses denoted as $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$. Furthermore, let there be $n$ students having enrolled in at least one of the $m$ courses in $\mathcal{C}$, which we denote as $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$. For each course $c_j$, we assume there are some course features that can be represented as the vector $f_j \in \mathbb{R}^{d_j}$ with $d_j$ being the dimension size after encoding the course features. Similarly for each of the students $s_i$ we assume there has been some collected demographic information that can be represented as the vector $d_i \in \mathbb{R}^{d_i}$ with $d_i$ being the dimension size after encoding the student demographic data. In addition to the demographic data, the system is assumed to have collected some sequential behavioral data for each student $s_i$ enrolled in course $c_j$ that we represent as $B_{ij} = [B_{ij}^1, B_{ij}^2, \ldots, B_{ij}^q]$ where $B_{ij}^q \in \mathbb{R}^q$ represents an encoding of the feature for student $s_i$ during the $i^{th}$ week of course $c_j$, $k$ represents the number of weeks for which behavioral data was collected, and $q$ is the dimension of the encoded weekly student behavior. In other words, we have a tensor of student behavioral data $B \in \mathbb{R}^{n \times m \times k \times q}$.

For each student $s_i$, we represent their performance outcome for course $c_j$ as $o_{ij}$, where we assume there can be $P$ outcomes (denoted by the set $p \in P$).

Now, given the notations listed above, we seek to learn a model $f(\cdot; \theta)$ having parameters $\theta$ such that it can predict the student course outcomes $O$ as follows:

$$M(\mathcal{C}, \mathcal{S}, F, D, B, O, f(\cdot; \theta)) \rightarrow \theta$$

where we use $M$ to denote the machine learning (artificial intelligence) process, $B$ is used to represent the behavioral (e.g., click) data for a given set of courses $\mathcal{C}$ using only the first $k$ weeks of data, $F$ represents the set of course features of $\mathcal{C}$, $D$ denotes the set of demographic data for the students in $\mathcal{S}$, and $O$ represents the performance outcomes of the students in $\mathcal{S}$ and the learned parameters of $f(\cdot; \theta)$ are given by $\theta$.

3. PROPOSED MODEL

In this section, we explain our proposed model in detail.

3.1 Knowledge Graph Representation

We first model the historical online course data in the form of a knowledge graph, as shown in Figure 2. Our knowledge graph formulation in Figure 2(b) offers a richer representation than a traditional independent naive student course relation representation shown in Figure 2(a). This is because through this graph structure we can leverage the relations between students and courses beyond that seen in Figure 2(a). We let $G = \{\mathcal{C}, \mathcal{S}, X_c, X_s, B, A\}$ represent a knowledge graph $G$ containing the set of $m$ course nodes $\mathcal{C}$, set of $n$ student nodes $\mathcal{S}$, course features $X_c \in \mathbb{R}^{n \times d_c}$ constructed from $F$, student demographic features $X_s \in \mathbb{R}^{m \times d_s}$ constructed from $D$, the behavioral data $B$ representing complex sequential edge features, and an adjacency tensor $A \in \mathbb{R}^{n \times m \times P}$ constructed from the $P$ different student-course outcome relations where $A_{ij}^p = 1$ if $o_{ij} \in O$ and $o_{ij} = p$ (with $A_{ij}^p = 0$ otherwise). Now, given the knowledge graph $G$, we seek to extract student and course embeddings by using a relational graph neural network.

3.2 Relational Graph Neural Network

Recently, graph neural networks (GNNs) [38, 39] have become increasingly popular due to their ability to utilize deep
learning on graph structure data. One popular class of GNNs is the graph convolutional networks (GCNs) \cite{3,27}, which are constructed with roots from the classical CNNs. The general idea of these GCN models is that we would like to learn a better set of latent features. In the context of our problem, to better understand and represent a student, rather than directly using their features alone, we could use a 1-layer GCN that would incorporate the features of all the courses that the student has taken. For example, in Figure 3(a), the 1-hop neighbors would be utilized in a 1-layer GCN model taking into consideration the course $c_3$ that they passed and $c_1$ that they failed. Then, it is natural to see in Figure 3(a) that using a 2-layer GCN would furthermore incorporate the 2-hop neighbors which would include information from all the classmates of $s_i$ for each of the two courses they have taken, and thus providing further context into learning a more comprehensive embedding for student $s_i$. We specifically harness the ability of a relational graph convolutional network \cite{34}. Next we will provide the details on how the first layer (or equivalently a 1-layer) GCN is able to construct learned representations $h_{s_i}^1$ and $h_{c_j}^1$ for the student $s_i$ and course $c_j$, respectively, from the initial student features $X_s$, course features $X_c$, and adjacency tensor $A$ in our knowledge graph representation.

First Layer Embeddings. First, we recall that connections between students and courses are stored in the tensor $A$ where $A_{ij} = 1$ if $o_{ij} \in O$ and $o_{ij} = p$ (with $A_{ij}^p = 0$ otherwise). Thus, we define for a student $s_i$ their set of courses for which they had outcome $p$ as $N^p_i(s_i)$. Similarly, we define for a course $c_j$ their set of students that received the outcome $p$ as $N^p_j(c_j)$. Now, given these new notations, we can define the first layer representations $h_{s_i}^1$ and $h_{c_j}^1$ for the student $s_i$ and course $c_j$, respectively, as follows:

$h_{s_i}^1 = \sigma \left( W_{s,1}^{ij} X_{s_i}^{[i]} + \sum_{p \in P} \frac{1}{|N^p_i(s_i)|} \sum_{c_j \in N^p_i(s_i)} W^p_j X_{c_j}^{[j]} \right) \quad (1)$

$h_{c_j}^1 = \sigma \left( W_{c,1}^{ij} X_{c_j}^{[j]} + \sum_{p \in P} \frac{1}{|N^p_j(c_j)|} \sum_{s_i \in N^p_j(c_j)} W^p_i X_{s_i}^{[i]} \right) \quad (2)$

where $\sigma$ is an element-wise non-linear activation function (e.g., ReLU($\cdot$) = max($0, \cdot$) \cite{13}), $X_{s_i}^{[i]}$ denotes the student features for $s_i$, $X_{c_j}^{[j]}$ denotes the course features for $c_j$, $W_{s,1}^{ij}$ is used to transform the self features from the original features, and $W_{c,1}^{ij}$ is used for transforming the features that are linked through the relation (i.e., course outcome type) $p$ for the first layer.

Final Student Embeddings. If we assume having $L$ layers in our GCN model, we can then define the last layer where we will obtain the student embedding $z_{s_i}^L = h_{s_i}^L$ for $s_i$ as follows:

$z_{s_i}^L = \sigma \left( W_{s,1}^{ij} h_{s_i}^{L-1} + \sum_{p \in P} \frac{1}{|N^p_i(s_i)|} \sum_{c_j \in N^p_i(s_i)} W^p_j h_{c_j}^{L-1} \right)$

$h_{s_i}^{L-1} = h_{s_i}^1$ from Eq. (1) and similarly $h_{c_j}^{L-1} = h_{c_j}^1$ from Eq. (4).

Final Course Embeddings. If we assume having $L$ layers in our GCN model, we can then define the last layer where we will obtain the course embedding $z_{c_j}^L = h_{c_j}^L$ for $c_j$ as follows:

$z_{c_j}^L = \sigma \left( W_{c,1}^{ij} h_{c_j}^{L-1} + \sum_{p \in P} \frac{1}{|N^p_j(c_j)|} \sum_{s_i \in N^p_j(c_j)} W^p_i h_{s_i}^{L-1} \right)$

where $h_{s_i}^1$ is the embedding of student $c_j$ at layer $l$ of the GCN. Next, we will discuss how DOPE uses an RNN to encode a student’s sequential behavior data associated with a given course.

3.3 Encoding Student Behavioral Data

In this part, we discuss how to encode the sequential edge features i.e., behavioural student data. To recall, when a student $s_i$ is currently enrolled in a course $c_j$, by the $k$-th week we will have the features $B_{ij} = [B_{ij}^{[1]}, B_{ij}^{[2]}, \ldots, B_{ij}^{[k]}]$. To better represent the behavioral data, we utilize a Long-Short Term Memory (LSTM) \cite{10}, which is an effective RNN variant that has been designed to extract temporal features from sequential data e.g., videos \cite{25}, speech \cite{14,25}, and text \cite{23,23}. Furthermore, it has shown great abilities to capture temporal online user behaviors \cite{26}. We fix the length of the behavior feature sequence for all students to be $k$ (e.g., 10 weeks). Then for a given behavioral sequential data $B_{ij}$, at each week $t \in [1, k]$, an LSTM unit takes the $t$-th week’s click feature vector $B_{ij}^t$ as the input and uses LSTM formulation \cite{10} to produce the output behavioral vector $h_{ji}^t$. The final output of the LSTM is $h_{ji}^k \in \mathbb{R}^h$ (i.e., output of last LSTM unit) when given the sequence $B_{ij}$ as input. Then, we set the encoded behavior of student $s_i$ for course $c_j$ as the $c^h$ dimensional vector $z_{ij}^h = h_{ji}^k$.

3.4 Final Course Performance Classifier

Here we combine student and course embeddings from the relational graph convolutional as well as encoded behavioral data and feed into a classifier. This can be seen in Figure 4. Given the student embedding $z_{s_i}^L$ for student $s_i$, course embedding $z_{c_j}^L$ for course $c_j$, encoded student behavior of $s_i$ in the course $c_j$ as $z_{ij}^h$, we form the final feature representation as follows:

$z_{ij} = z_{s_i}^L \| z_{ij}^h \| z_{c_j}^L$

where $\|$ denotes concatenation and we concatenate the three components together into a single $(c^L + c^h + c^L)$ dimensional representation. For training DOPE, we use supervised learning such that labels are the outcome performances from the historical data $o_{ij} \in O$ and matched with the training student and course pair $(s_i, c_j)$. More specifically, we construct a minibatch set $M$ that contains triplets of the form $(s_i, c_j, T)$ where $T = o_{ij}$ (i.e., the course outcome type) and we assume the outcome type set $T$ where $|T| = p$ since there are $p$ course outcome types. The objective is then formalized in the following:

$L = \frac{1}{|M|} \sum_{(s_i, c_j, T= o_{ij}) \in M} \log \frac{1}{\sum_{T \in T} \exp \left( \theta_{MLG} z_{ij}^T \right) + \lambda \text{Reg}(\theta_{RGCN}, \theta_{LSTM}, \theta_{MLG})}$

where the classifier first maps $z_{ij}$ to a $p$ dimensional vector through the parameters $\theta_{MLG}$ (since we have $p$ different outcomes, i.e.,
We compare DOPE with the baseline methods for the different numbers of weekly click data i.e., 5, 10, 15, and 20 weeks. By doing so, we can measure how effective DOPE is in the early prediction of a student’s course performance prediction. We note that 20 weeks is almost half of a course period when there is still adequate time for intervention in the case of prediction as failure.

4.3 Binary Classification
As mentioned before, our dataset includes 4 distinct labels for a student’s performance in a course, namely Distinction, Pass, Fail, and Withdrawn. In this section, we merge Distinction and Pass into a single class “Pass” and Fail and Withdrawn into a single class “Fail” and then perform a binary classification. Figure 5 illustrates the experimental results for all courses. We make the following observations based on the results presented in Figure 5:

- In general, the more weekly click data is introduced, the better we can predict the students’ outcomes. DOPE enjoys more of such performance increase as compared to other methods. In particular, as early as 20 weeks from the start of a course (i.e., almost in the middle of a course duration), it can predict student’s outcomes with very high performance. This allows teachers or online course administration to take actionable and interventional measures to help students with poor performance.
- DOPE achieves a better performance than DOPEFCN. This shows the fact the LSTM component as a machinery extracting temporal features from click behaviors is necessary and affects the model’s predictive power.
- DOPE is shown to be effective for all courses as we can observe it achieves an F1 score of more than 0.8 across all courses when 20 weeks of click data are considered.

4.4 4-class Classification
In this part, we compare the performance of DOPE with baseline methods for a 4-class classification setting whose experimental results are demonstrated in Figure 6. We make the following observations based on the results presented in Figure 6:

- The observations we made for binary classification hold for 4-class classifications as well. In particular, DOPE still outperforms baseline approaches, more weekly click data is helpful in course outcome prediction, and the LSTM can effectively handle sequential that than simple concatenation followed by a fully connected network model (i.e., DOPEFCN).
- Since more classes are considered, compared to binary classification, the 4-class classification is a harder task. In particular, now Withdrawn is considered as a separate class, which might be “conceptually” hard for a model to discern from Fail.

4.5 Behavioral Feature Analysis
Since behavioral data (i.e., click data) plays an essential role in determining a student’s performance, we conduct a feature analysis experiment investigating the importance of each behavior type. A similar feature analysis has been performed to discover great insights into human behaviors [21]. To this end, we follow an ablation feature analysis where at each time we include one feature type and suppress the rest (setting their values to zero) and then acquire the F1 score from the model. We do this experiment for the binary classification and the case when 20 weeks of click data is included. Figure 6 demonstrates the results and we make the following observations accordingly:

- For all courses, feature type homepage is associated with a high F1 score. This seems reasonable since most of the click activity occurs on the main page of the platform interface.
- Interestingly, clicks and activities in forums have an influential role in predicting fail or pass of a student in a course. This is in line with previous [19,23] where they showed that MOOC forum activities correlate with a student’s academic performance.

<table>
<thead>
<tr>
<th>Name</th>
<th>Train Periods</th>
<th>Test Periods</th>
<th>#Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS1</td>
<td>2013J</td>
<td>2014J</td>
<td>735</td>
</tr>
<tr>
<td>SS3</td>
<td>2013J, 2014B</td>
<td>2014J</td>
<td>2366</td>
</tr>
<tr>
<td>ST2</td>
<td>2013J, 2014B</td>
<td>2014J</td>
<td>2685</td>
</tr>
</tbody>
</table>

Table 1: The description of the dataset.
Unique behavioral importance profile of a course can aid policymakers, administrators and even course interface designers to prepare the course materials in a more informed way. For instance, we can observe that attribute wiki is playing an important role in performance prediction of ST2 while its effect is negligible for other courses. This can be indicative of materials of the course ST2 to be requiring more wiki access and consequently, the content can be changed accordingly.

Based on the observations above, we can conclude that DOPE encodes behavioral data in an intuitive manner that conforms to previous studies’ findings as well.

### 4.6 Inter-course Outcome Evaluations

Naturally, each course has its own model. However, in this section, we intend to measure inter-course performance evaluation where we train DOPE on one course and test it on another one. Table 2 shows the results. Again, the models are trained for the binary classification and they incorporate 20 weeks of the click data. Also, for the reference, we have included intra-course performance (i.e., the same course for training and test) shown in the diagonal entries of Table 2. Expectedly, when the training course and the test course are the same (i.e., intra-course setting), the performance is higher. This seems reasonable since clicking patterns are expected for the course in the past (i.e., a part of the training data) and the one in the future (i.e., testing data), and the model can more easily extract such patterns. Although the results for inter-course results are not as good as the ones for intra-course, we still see that the DOPE can effectively achieve reasonable performance. This indicates that the proposed model DOPE can detect salient click and demographic patterns that are transferable from a course to another.

<table>
<thead>
<tr>
<th>Training course</th>
<th>Test course</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS1</td>
<td>SS2</td>
</tr>
<tr>
<td>SS1</td>
<td>0.83</td>
</tr>
<tr>
<td>SS2</td>
<td>0.63</td>
</tr>
<tr>
<td>SS3</td>
<td>0.64</td>
</tr>
<tr>
<td>ST1</td>
<td>0.60</td>
</tr>
<tr>
<td>ST2</td>
<td>0.74</td>
</tr>
<tr>
<td>ST3</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Table 2: Inter-course performance evaluation
5. RELATED WORK

In the following, we highlight some of the works focusing on student dropout and performance prediction. In [35], they extracted 27 interpretive features and used logistic regression to predict student persistence prediction. The authors of [32] used probabilistic soft logic to model student survival by constructing probabilistic soft logic rules and associating them. Different from [32] (which mainly considered forum features), in [28] they did not consider forum data, but instead only made use of clickstream data to train their prediction model. More specifically, they used principal component analysis [37] paired with a linear support vector machine [4] for each week. It was in [12] that a more comprehensive approach was taken that used standard classification trees [2] and adaptive boosted trees [1] to construct their two-stage Friedman and Nemenyi procedure for dropout prediction by processing different features such as clickstream-based, forum-based, and assignment-based features. More recently, in [3], the authors studied a hybrid method for dropout prediction by combining both a decision tree [11] and extreme learning machine [18]. In addition to these traditional machine learning methods, some researchers have tried to use different deep learning models for dropout prediction of online courses. In [9] an LSTM was used to deal with the features extracted from students’ interaction with lecture videos, forums, quizzes, and problems. [36] explored the potential benefits of employing a fully connected feed-forward neural network for dropout prediction. Different from previous work, [10] proposed a context-aware feature interaction network to incorporate context information of both participants and courses. More specifically, they used an attention-based mechanism for learning activity features. The most similar method to ours is found in [17] where they sought to conduct performance evaluations on students using a graph neural network (GNN), but there are primary differences: (1) they constructed separate small graphs of courses for each student while DOPE constructs a single knowledge graph of historical student course relations; (2) their graph neural network was used to obtain a graph classification for a given student based on that student’s specific course graph, while our method uses the relational graph neural network to learn embeddings for both students and courses from a single large knowledge graph; and (3) DOPE furthermore utilizes an LSTM model to capture a student’s rich sequential behavioral data beyond just using static fixed student features.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a model for course performance prediction we call it Deep Online Performance Evaluation (DOPE). Our method first represents the online learning system as a knowledge graph, such that we then learn student and course embeddings from historical data using a relational graph neural network. Simultaneously, DOPE utilizes an LSTM for harnessing the student behavior data into a condensed encoding, as the data has a natural inherent sequential form. We tested the proposed model on six courses from the OULAD dataset where the results showed the feasibility of DOPE and that it can predict at-risk students of on-going courses. We also investigated the usefulness of the different types of behavioral features and observed that DOPE encodes the data in an intuitive manner.

In the future, we will first analyze the imbalance and sparse issues of the dataset. One possible way to alleviate the sparsity would be through a network alignment [6] of multiple MOOC datasets represented as knowledge graphs or connecting student behavior data from social media for better predictions in online education [22]. Also, we will investigate more advanced ways of handling behavioral data. For example, investigating better ways to use “subpage” clicks beyond a simple aggregation that ignores separating the multiple different “subpages”. In addition, we plan to apply our framework to the traditional education system aiming at identifying similarities and differences between online and traditional course performance prediction, since we believe this to be highly important in improving online learning systems.

Acknowledgement

This research is supported by the National Science Foundation (NSF) under grant numbers IIS1907704, IIS1928278, IIS1714741, IIS1715940, IIS1845081, and CNS1815636.
7. REFERENCES


